Complex Cross Ambiguity Function Post-Decomposition Spoofing Detection with Inverse RAIM

Sahil Ahmed, Samer Khanafseh, Boris Pervan, *Illinois Institute of Technology*

Biographies

Sahil Ahmed is currently a Ph.D. Candidate at the Navigation Laboratory in the Department of Mechanical and Aerospace Engineering, Illinois Institute of Technology (IIT). He also works as a Pre-Doctoral Researcher at Argonne National Laboratory. His research interests include Spoofing Detection in GNSS receivers, Software-Defined Radios (SDR), Satellite Communication, Statistical Signal Processing, Estimation and Tracking, Sensor Fusion for autonomous systems, Sense and Avoid algorithms for Unmanned Aerial Vehicles (UAV), Machine learning & Deep Learning Algorithms.

Dr. Samer Khanafseh is currently a research associate professor at Illinois Institute of Technology (IIT), Chicago. He received his PhD degrees in Aerospace Engineering from IIT in 2008. Dr. Khanafseh has been involved in several aviation applications such as Autonomous Airborne Refueling (AAR) of unmanned air vehicles, autonomous shipboard landing for the NUCAS and JPALS programs, and the Ground Based Augmentation System (GBAS). His research interests are focused on high accuracy and high integrity navigation algorithms, cycle ambiguity resolution, high integrity applications, fault monitoring, and robust estimation techniques. He is an associate editor of IEEE Transactions on Aerospace and Electronic Systems and was the recipient of the 2011 Institute of Navigation Early Achievement Award for his outstanding contributions to the integrity of carrier phase navigation systems.

Dr. Boris Pervan is a Professor of Mechanical and Aerospace Engineering at the Illinois Institute of Technology (IIT), where he conducts research on high integrity navigation systems. Prior to joining the faculty at IIT, he was a spacecraft mission analyst at Hughes Aircraft Company (now Boeing) and a postdoctoral research associate at Stanford University. Prof. Pervan received his B.S. from the University of Notre Dame, M.S. from the California Institute of Technology, and Ph.D. from Stanford University. He has received the IIT Sigma Xi Excellence in University Research Award (twice), IIT University Excellence in Teaching Award, IEEE Aerospace and Electronic Systems Society M. Barry Carlton Award, RTCA William E. Jackson Award, Guggenheim Fellowship (Caltech), and the Albert J. Zahm Prize in Aeronautics (Notre Dame). He is a Fellow of the Institute of Navigation (ION) and former Editor-in-Chief of the ION journal *NAVIGATION*.

Abstract

In this paper, we present decomposition results of the Complex Cross Ambiguity Function (CCAF) of spoofed Global Navigation Satellite System (GNSS) signals into their constitutive components [1]. We also propose a new, post-decomposition detection algorithm based on a new "inverse" Receiver Autonomous Integrity Monitoring (RAIM) concept. The goal is to differentiate the spoofed and the authentic satellite signals to generate an authentic navigation solution. First, each satellite provides the two sets of signal parameters (code phases) post-decomposition. Using combinations of these sets, we calculate the pseudorange residuals and identify the two consistent (the authentic and spoofed) navigation solutions among all possible signal combinations over different times. The method is applicable to spoofing scenarios that can lead to Hazardous Misleading Information (HMI) and are difficult to detect by other means. The method can identify spoofing in the presence of multipath and when the spoofing signal power matches with offsets in code delay and Doppler frequency relatively close to the true signal. Spoofing can be identified at an early stage within the receiver without additional augmented sensors.

INTRODUCTION

Global Navigation Satellite Systems (GNSS) are the foundation of modern technological infrastructure. GNSS is used for Positioning, Navigation, and Timing (PNT) worldwide with applications in aviation, automated vehicle systems, telecommunication, finance, and energy systems. GNSS signals are vulnerable to Radio Frequency Interference (RFI) such as jamming and spoofing attacks. Jamming can deny access to GNSS service while spoofing can create false positioning and timing estimates that can lead to catastrophic results. This paper focuses on the detection of intentional RFI known as spoofing, a targeted attack where a malicious actor takes control of the victim's position and/or time solution by broadcasting counterfeit GNSS signals [2]. Different methods have been proposed to detect spoofing, such as received power monitoring which monitors

the response of automatic gain control (AGC) and can be used when an overpowered spoofing signal is broadcast, signal quality monitoring (SQM) which tracks the distortion of the autocorrelation function using I and Q channels, RAIM checks on inconsistent sets of five or more pseudoranges that allow the receiver to detect spoofing with one or more false signals, signal direction of arrival (DoA) estimation techniques using directional antennas or moving antennas in a specified pattern to observe if all satellite signals are broadcast from the same direction, inertial navigation system (INS) aiding [3] [4] which is based on drift monitoring, and others [5] [6]. Each of these methods have their own advantages and drawbacks. CAF (Cross Ambiguity Function) monitoring approaches [7], which exploit only the magnitude of the Complex CAF (CCAF), can be used to detect spoofing but face difficulties in environments with multipath and when the Doppler frequency and code phase of the received signal are closely aligned with the spoofed signal. There are machine learning and deep learning approaches (for example convolutional neural networks) to detect GNSS spoofing attacks using CAF, but these methods depend upon the availability of spoofing data and are limited to the dataset upon which they are trained [8]. A sampled signal can be represented in the form of a complex number, I (in-phase) and Q (quadrature), as a function of code delay and Doppler offset. In previous CAF monitoring concepts, a receiver performs a two-dimensional sweep to calculate the CAF by correlating the received signal with a locally generated carrier modulated by pseudorandom code for different possible code delay and Doppler pairs. Spoofing is detectable when two peaks in the CAF are distinguishable in the search space. This could happen, for example, if a power matched spoofed signal does not accurately align the Doppler and code phase with the true received signal. In practice, because detection using the CAF is not reliable under multipath or if the spoofed signals are close to the true ones, we instead exploit the full CCAF.

We implemented a method to decompose a CCAF made up of N contributing signals by minimizing a least-squares cost function. Because the optimization problem is non-convex, we implemented a Particle Swarm Optimization (PSO) algorithm to find the global minimum. The algorithm can decompose a sum of GNSS signals for a given satellite (e.g., true, spoofed, and multipath) into its respective defining parameters—signal amplitudes, Doppler frequencies, code delays, and carrier phases. The same process is performed for each visible satellite, and the estimated code phases are then used in the next step, which is the detection function.

The spoofing detection concept is as follows: consider a signal associated with a given satellite with three extracted code phases, associated with the true, spoofed, and multipath component. At first it is unknown which code phase corresponds to either authentic signal or spoofed signal. Now consider a set of redundant satellites (e.g., five or more for single constellation GNSS, six or more for dual constellations, etc.). The true code phases will be consistent, in a RAIM sense, across all the satellites. The same would be true for the spoofed code phases. But the multipath code phases would not be self-consistent. Therefore, we may assert that spoofing is happening if more than one set of code phases passes a RAIM test. The process is termed "Inverse RAIM" because detection is based on an extra set "passing" the RAIM test. The new algorithm is validated in simulation and against publicly available spoofing datasets, including TEXBAT [9].

GNSS Signals

GPS paved the way for global satellite navigation, followed by GLONASS, Galileo, BeiDou, and regional systems such as IRNSS and QZSS. These satellite constellations have different frequencies and signal structures for both civil and military use. Current and planned signals transmit on L1 (1575.42 MHz), L2 (1227.60 MHz), and modernized L5 (1176.45 MHz). Techniques such as code division multiple access (CDMA) and frequency division multiple access (FDMA) are used to distinguish different satellites in the constellation. Many modern signals also transmit a pilot component along with the data component of the signal which are not modulated with any navigation data and can be used to enhance the receiver signal processing capability. Data is modulated on the signals either through binary phase shift keying (BPSK) or binary offset carrier (BOC). Usually, all GNSS signals have a chip (rectangular) waveform. In BPSK, the 0s and 1s in a binary message are represented by two different phase states in the rectangular waveform. BOC modulation divides each chip into subchips represented as BOC (f_s, f_c) where f_s is the subcarrier frequency and f_c is the carrier frequency. BOC also divides the power spectrum main lobe into two identical lobes around the carrier frequency. We are using the GPS L1 C/A signal as an example in this work, but this spoofing detection method is applicable to all GNSS constellations and frequencies. The GPS L1 C/A signal is transmitted at a frequency of $f_L = 1575.42$ MHz (approximately 19 cm wavelength) from all satellites in the form of radio waves that are modulated with a pseudo-random (PRN) code $x(t)$ at the rate of 1.023 Mega-chips per second (300 m chip length) to distinguish between different satellites, and then again modulated with Navigation Data $D(t)$ at the rate of 50 bits per second.

GPS Receiver Architecture

As shown in Figure 1, the GPS signal is received at a receiver's antenna with code delay τ, Doppler f_D, and carrier phase θ. The signal is then amplified, passed through a band pass filter, and then down converted to an intermediate frequency f_{IF} by mixing with a locally generated mixing signal. It is then passed through a low pass filter to remove the high frequency components. The advantage of converting the signal to an intermediate frequency is that it simplifies the subsequent stages, making filters easier to design and tune. The signal is then digitized and mixed again (in Figure 2) with two locally generated replicas of the carrier signal $\bar{f}_D$, in-phase and quadrature, differing in phase by a quarter cycle, $\bar{\theta}$ and $\bar{\theta} + \frac{\pi}{2}$. During digitization, the signal is sampled at a sampling frequency based on the Nyquist rate to reliably capture the signal form. It is then passed through a low pass filter to remove the intermediate frequency, and finally it is mixed with a local replica of the PRN code with delay $\bar{\tau}$.

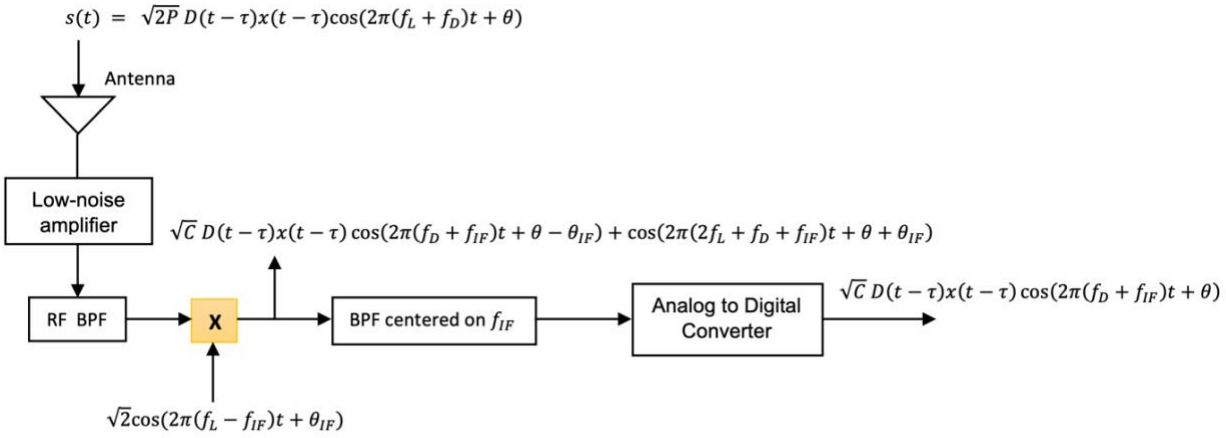


Figure 1. The front end of a GPS receiver.

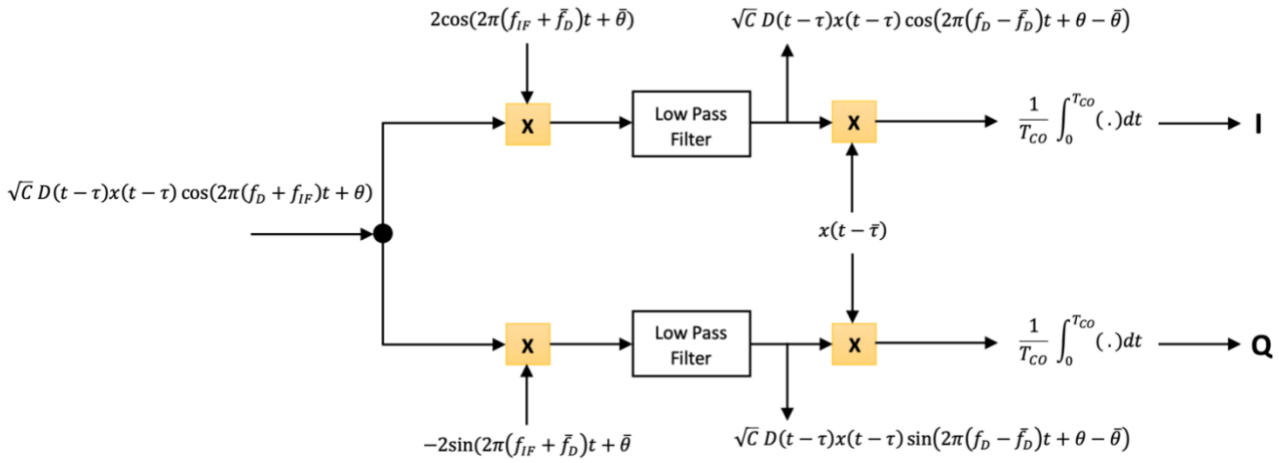


Figure 2. GPS receiver architecture after signal is digitized.

In-phase and Quadrature components

The in-phase I and quadrature Q components of an uncorrupted output signal (i.e., no spoofing or multipath) with amplitude $\sqrt{C}$ are shown in Equations (1) and (2). When presented in complex form, as in Equation (3), the in-phase and quadrature components are the real and imaginary parts of the signal, respectively. The coherent integration time T_{CO} can range from 1 to 20 milliseconds, with the upper limit designed to avoid integration across boundaries of a GPS data bit $D(t)$ shown in Figure 3. Coherent integration is performed to reduce the effects of thermal noise. Longer coherent integration times may also be limited by satellite Doppler, receiver oscillator error and drift, and receiver motion.

$$I\left(\sqrt{C},\tau,f_D,\theta;\bar{\tau},\bar{f}_D,\bar{\theta}\right)=\frac{\sqrt{C}}{T_{CO}}\int_0^{T_{CO}}x(t-\tau)x(t-\bar{\tau})\cos\left(2\pi\left(f_D-\bar{f}_D\right)t+\theta-\bar{\theta}\right)dt \tag{1}$$

$$Q\left(\sqrt{C},\tau,f_D,\theta;\bar{\tau},\bar{f}_D,\bar{\theta}\right)=\frac{\sqrt{C}}{T_{CO}}\int_0^{T_{CO}}x(t-\tau)x(t-\bar{\tau})\sin\left(2\pi\left(f_D-\bar{f}_D\right)t+\theta-\bar{\theta}\right)dt \tag{2}$$

$$S=I+iQ \tag{3}$$

Performing the integrals in Equations (1) and (2), Equation (3) can be expressed as (4) (details shown in the appendix)

$$S\left(\sqrt{C},\tau,f_D,\theta;\bar{\tau},\bar{f}_D,\bar{\theta}\right)=\sqrt{C}\,R(\tau-\bar{\tau})\,\mathrm{sinc}\left(\pi\left(f_D-\bar{f}_D\right)T_{CO}\right)\exp\left(i\pi\left(\left(f_D-\bar{f}_D\right)T_{CO}+\theta-\bar{\theta}\right)\right) \tag{4}$$

where

$$R(\xi)=\begin{cases}\dfrac{\xi}{T_c}+1 & -T_c<\xi<0\\ \dfrac{-\xi}{T_c}+1 & 0<\xi<T_c\\ 0 & \text{otherwise}\end{cases} \tag{5}$$

and T_c is the duration of a single chip.

To simplify the notation, we define $a\triangleq\sqrt{C}$. Summing N component signals (for example, assuming a true satellite signal, a spoofed signal, and a single multipath signal, $N=3$), we have[+]

$$S_N(g|\bar{\tau},\bar{f}_D,\bar{\theta})=\sum_{j=1}^{N}a_j\,R\left(\tau_j-\bar{\tau}\right)\mathrm{sinc}\left(\pi\left(f_{D_j}-\bar{f}_D\right)T_{CO}\right)\exp\left(i\pi\left(\left(f_{D_j}-\bar{f}_D\right)T_{CO}+\theta_j-\bar{\theta}\right)\right) \tag{6}$$

where $g=\left(a_1,\tau_1,f_{D_1},\theta_1,\ldots,a_N,\tau_N,f_{D_N},\theta_N\right)$.

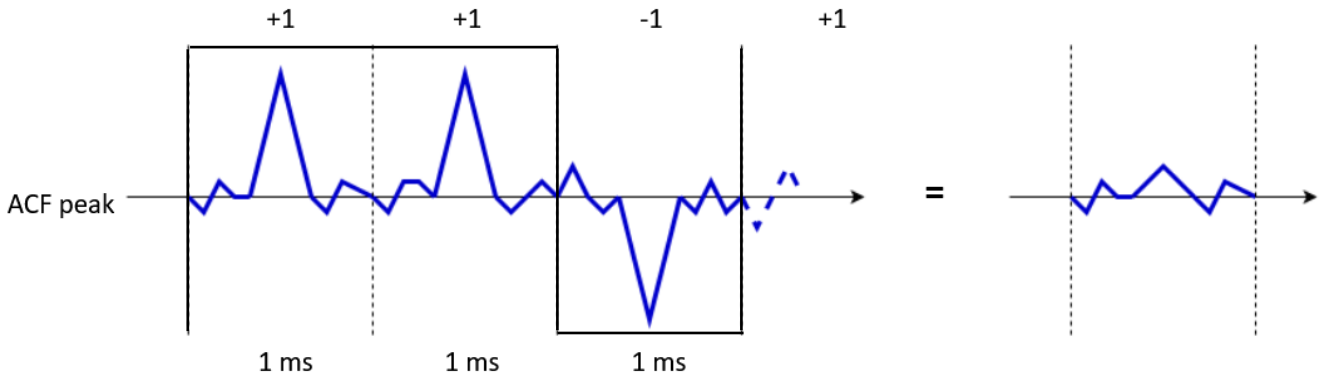


Figure 3. Phase change at navigation data bit boundary which limits the coherent integration time.

[+] Strictly speaking, Equation (6) is true only for infinite length random codes. For finite length PRN codes like GPS L1 C/A, $R(\xi)$ will have additional small, but non-zero, values outside the domain $\xi\in(-T_c,T_c)$. We ignore these for now but will address their impact later.

Complex Cross Ambiguity Function (CCAF) Measurement Space

The Complex Cross Ambiguity Function (CCAF) measurements discretely span the code delay ($\bar{\tau}$) and Doppler frequency ($\bar{f}_D$) space. At present, to limit the size of the measurement data, we set $\bar{\theta} = 0$. The upper limit on the code delay dimension is the length of the code itself and Doppler frequency dimension usually well within $\pm$4000 Hz. In the absence of spoofing and multipath (and noise), the CCAF measurement landscape looks like Figure 4.

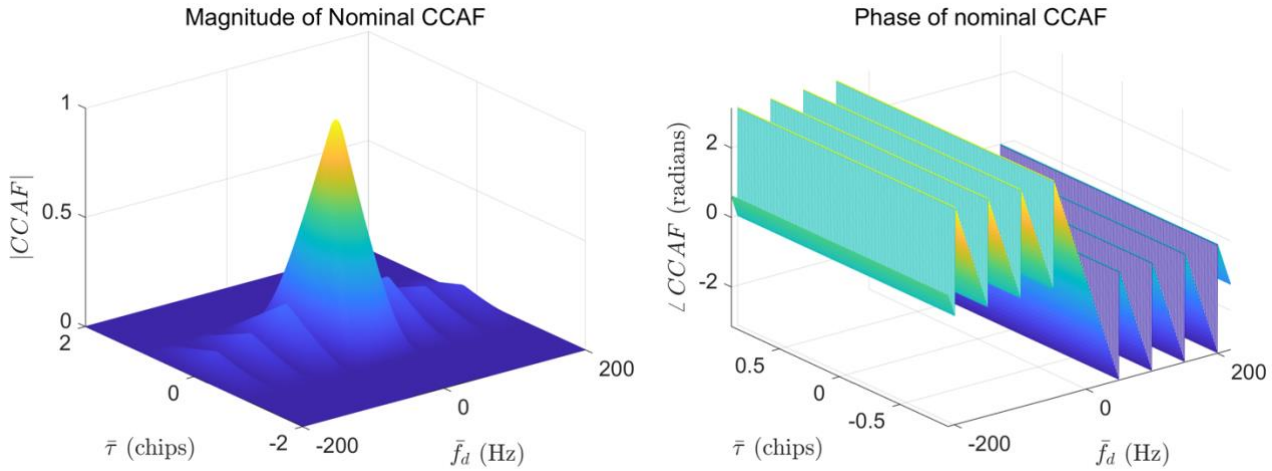


Figure 4. Magnitudes (left) and phases (right) of CCAF measurements of CCAFs when only the authentic signal is present.

When visualized from the code delay point-of-view, the magnitude is a triangle with base length of 2 chips as shown in Figure 5 (left) and the CCAF phase change happens at correlation peak (Figure 5 (right)). From the Doppler frequency point-of-view, the magnitude of CCAF is represented by a sinc function (Figure 6 (left)) and the phase of CCAF by a sawtooth pattern (Figure 6 (right)) with frequency $1/T_{CO}$. The software defined radio [10] allows flexibility to arbitrarily change Doppler spacing. However, the code delay spacing is limited by the sampling rate of the receiver.

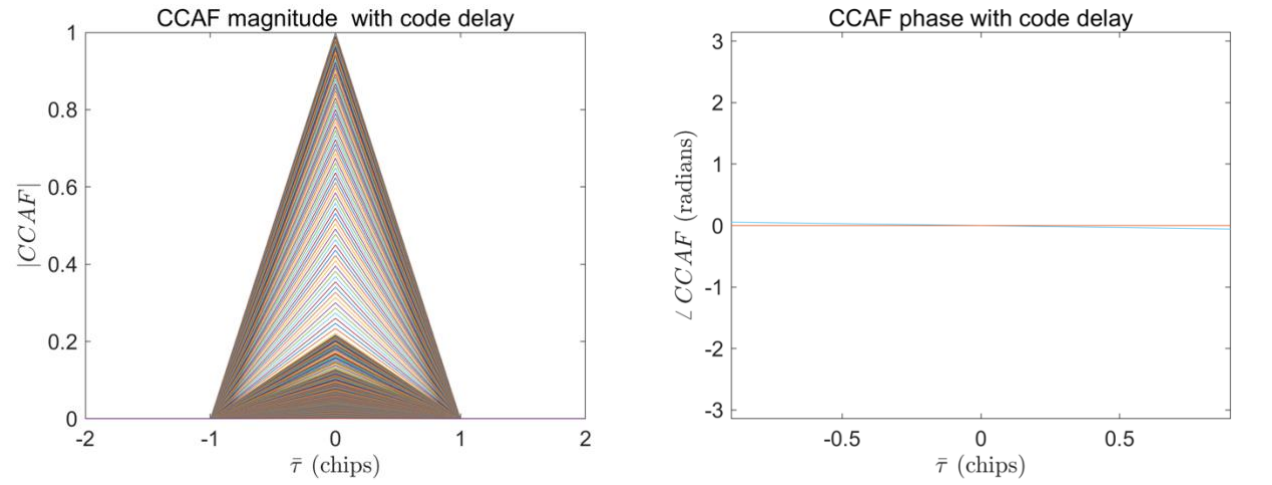


Figure 5. Magnitudes (left) and phases (right) of CCAF measurements from code delay point-of-view.

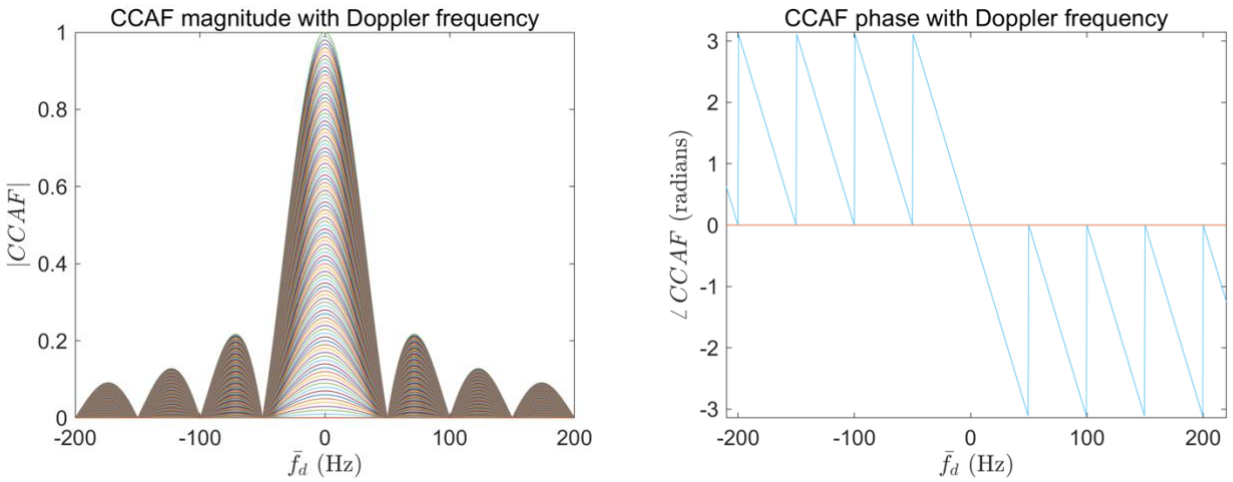


Figure 6. Magnitudes (left) and phases (right) of CCAF measurements from Doppler point-of-view.

SPOOFING

GNSS spoofing techniques consist of broadcasting fake GNSS signals with the goal of taking control of a GNSS receiver and introducing false results for positioning or timing or both. A spoofing attack can be very sophisticated by replicating and transmitting the signal parameters (amplitude, code phase, and Doppler) relatively close to the authentic signal parameters. However, it is very hard to replicate the precision of carrier phase, and we want to exploit this by observing the CCAF. When a spoofer initiates a subtle spoofing attack, it generates a signal with the same code phase and Doppler frequency pair as the authentic signal, and then slowly pulls away the code phase/Doppler frequency. A chip is 300 m in length (for the GPS L1 signal), and a change in a fraction of a chip can lead to a significant change in the PNT solution. Newer L5 signals have a faster chipping rate, and one chip length is 30 meters. We are focusing on scenarios where the spoofing signals are in the vicinity of ± 1 chip.

When a spoofed signal is present and the code delays and Doppler frequencies of the signals are not closely aligned, two peaks are visible in the magnitude of the CCAF, as shown in Figure 7 (left), and the phase, in Figure 7 (right), is considerably distorted relative to the unspoofed case in Figure 4 (right). The two peaks merge if the code delays and Doppler frequencies are closely aligned, as shown in Figure 8 (left). However, the phase in Figure 8 (right) is still significantly different from the unspoofed case.

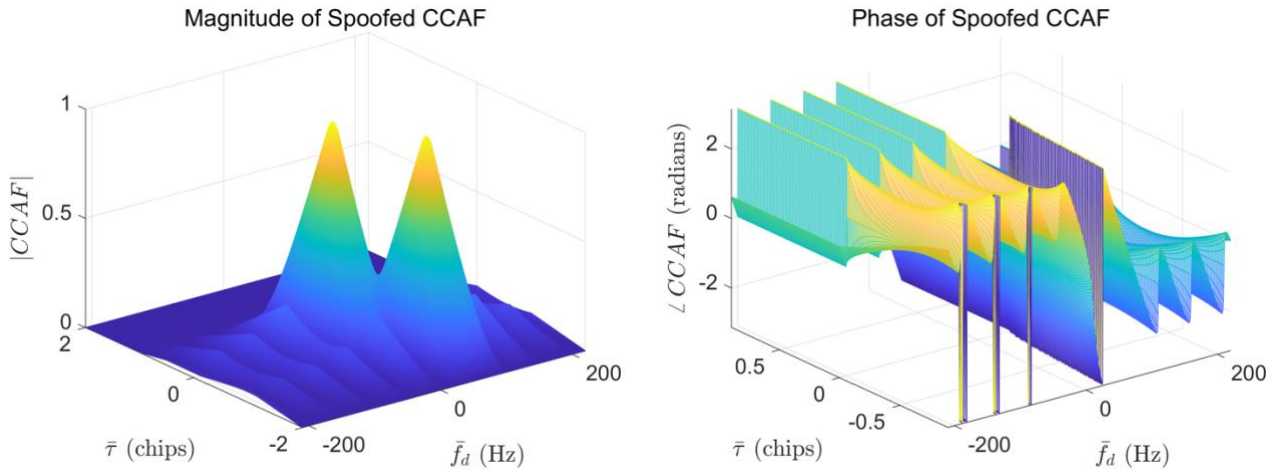


Figure 7. Magnitudes (left) and phases (right) of CCAF measurements when code delay and Doppler frequency pairs are far apart.

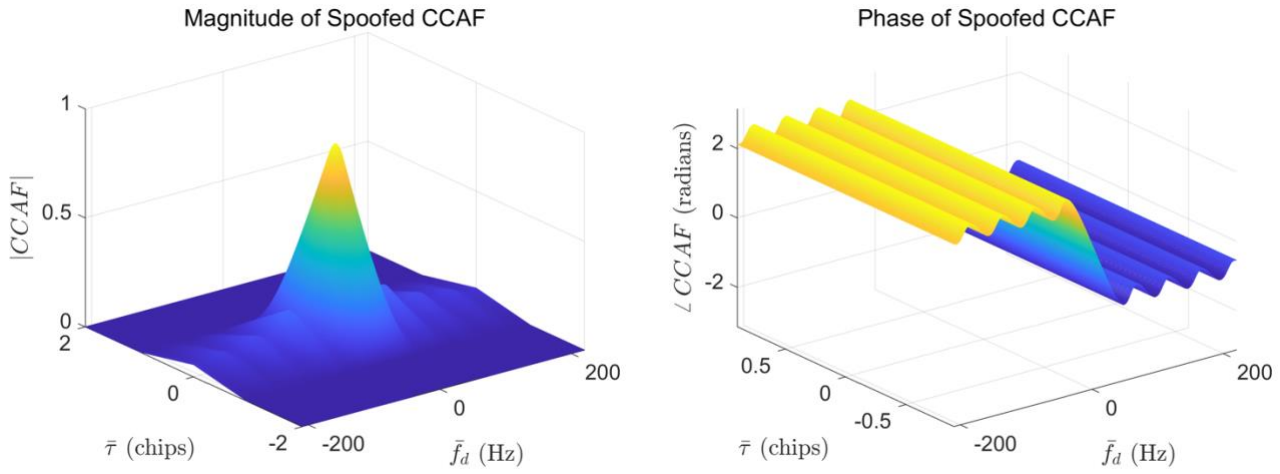


Figure 8. Magnitudes (left) and phases (right) of CCAF measurements when code delay and Doppler frequency pairs are closely aligned.

PARTICLE SWARM DECOMPOSITION

Stacking the CCAF measurements from the grid space $(\bar{\tau}, \bar{f}_D)$, the measurement model can be written as

$$z = S_N(g|\bar{\tau},\bar{f}_D) + \nu \tag{7}$$

where ν is the vector of measurement errors, including the effects of thermal noise and code cross-correlation. To decompose the N signals, we seek to obtain an estimate of the parameter vector, $\hat{g}$, that minimizes the cost function

$$J = \left\| z - S_N(\hat{g}|\bar{\tau},\bar{f}_D) \right\|^2. \tag{8}$$

Unfortunately, due to the structure of S_N the cost function is non-convex, and a global minimum cannot be obtained by standard gradient-based methods. In computational science, Particle Swarm Optimization (PSO) [11]is an optimization algorithm that works by generating a population of "particles" randomly which are actually candidate solutions given upper and lower bounds. A simple PSO algorithm is shown in Figure 9. The particles are moved around in the N dimensional space based on their own best-known position p_i and entire population's best-known position b as shown in Equations (9) and (10). When a particle finds a position/solution that minimizes the cost function better than the previous known position, p_i gets updated based on Equation (11). If that particle's position is best among all other particles' positions (minimizes the cost function), b is updated based on Equation (12) and called the best global solution of the swarm.

PSO Algorithm

Generate n particles randomly with "position" $x_i(t) \in \boldsymbol{X}$ and "velocity" : $v_i(t) \in \boldsymbol{V}$

For each $i = 1,2, \ldots \ldots, n$ particle:

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{9}$$

$$v_i(t+1) = w * v_i(t) + c_1 * r_1 * \big(p_i(t) - x_i(t)\big) + c_2 * r_2 * (b(t) - x_i(t)) \tag{10}$$

$$p_i(t+1) = \begin{cases} p_i(t) & f\big(p_i(t)\big) \le f\big(x_i(t+1)\big) \\ x_i(t+1) & f\big(p_i(t)\big) > f\big(x_i(t+1)\big) \end{cases} \tag{11}$$

$$b(t+1) = \max\big\{f\big(p_i(t)\big), f\big(b(t)\big)\big\} \tag{12}$$

where:

r_1, r_2	are the uniformly distributed random numbers with $\mathcal{N}(\mu, \sigma^2)$
w	is the inertia coefficient
c_1, c_2	are the acceleration coefficient
$p_i(t)$	is the best local position
$b(t)$	is the best global position

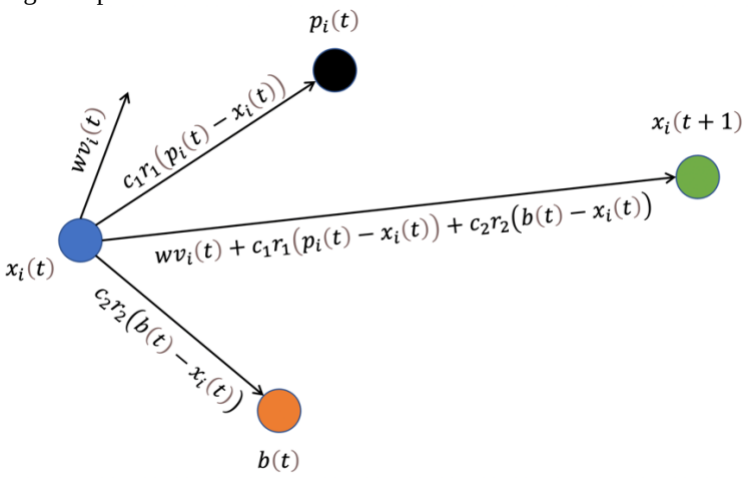


Figure 9. Search mechanism of the particle swarm optimization algorithm with particle position updates based on hyperparameters.

The PSO algorithm is applied to minimize the cost function J in Equation (8). As the measurement vector z may be comprised of N signals, the parameter vector $\hat{g} = \left(\hat{a}_1, \hat{\tau}_1, \hat{f}_{D_1}, \hat{\theta}_1, \ldots, \hat{a}_N, \hat{\tau}_N, \hat{f}_{D_N}, \hat{\theta}_N\right)$ that yields the best global solution defines our CCAF decomposition.

RESULTS

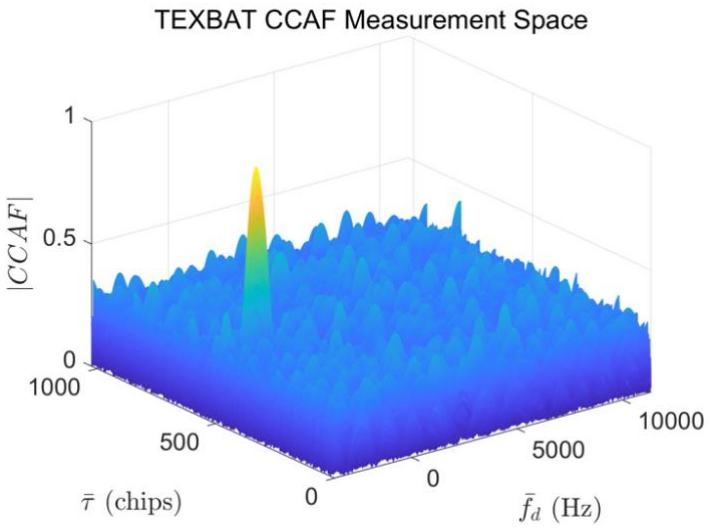
TEXBAT Dataset

In [1] we showed the capability of the PSO algorithm to decompose CCAF made of up to N contributing signals and output the parameter vector $\hat{g}$ when noise and code cross-correlation are not present. To test the algorithm in a more realistic scenario, we take a section of the TEXBAT dataset scenario 4, *power-matched position push with power advantage of 0.4 dB,* that includes thermal noise and cross correlations. The measurement space for PRN 13 consists of 1023 chips that are distributed over 25,000 samples, i.e., code delay bins, with Doppler frequency bin widths of 10 Hz, and a total of 1501 bins. This can be seen in the figure in Case 1 (right), where two signals are present. The PSO searches for three signals, while the input CCAF has two prominent signals present. As shown in the Case 1 table, the algorithm correctly decomposes the signal parameters. The two signals detected by the algorithm are the authentic signal and the spoofing signal in the measurement space, while the third signal is estimated to have almost zero amplitude. The two detected signals are zoomed in and shown in Figure 10.

The noise floor as shown in the Case 1 CCAF measurement space includes thermal noise, and cross correlations are evident. The noise floor is reduced by increasing the coherent integration time. In Case 2, the coherent integration time is 20 milliseconds, since for GPS L1 C/A signal the Navigation Data bit is 20 milliseconds long. In Case 2, the peak is reduced in width as the sinc function has a frequency of $1/T_{CO}$. In both Case 1 and Case 2, output parameters are relatively close to each other, but we don't have any information about the true parameters. The third signal output by the particle swarm decomposition algorithm has zero amplitude indicating that there is no third signal present.

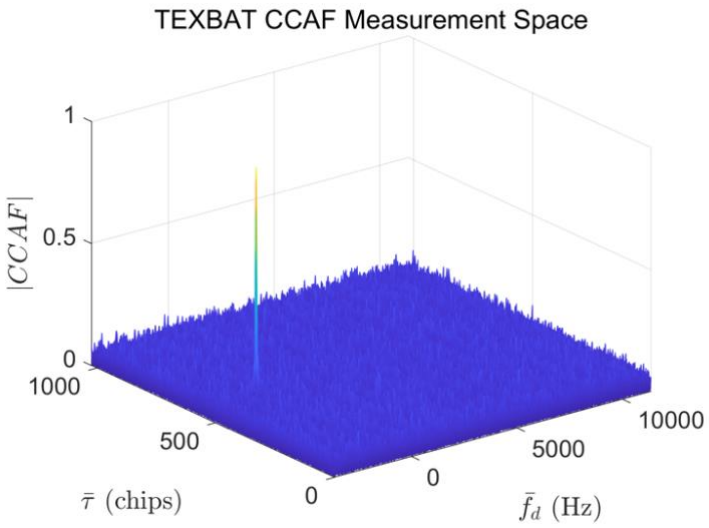
The zoomed-in view of both Case 1 and Case 2 along a constant Doppler cut is shown in Figure 10. The noise floor with 20 ms coherent integration time (Figure 10 (right)) is significantly lower than the 1 ms coherent integration time (Figure 10 (left)). Both results are normalized with one of the peaks representing an authentic signal and the other representing a spoofing signal. The peaks' magnitudes also change with respect to each other when the coherent integration time changes from 1 ms to 20 ms.

CASE 1	Output Parameters
	$\hat{g}$
a_1	0.9891
τ_1 (chips)	507.7127
f_{D_1}(Hz)	-1669.0176
θ_1 (rad)	-0.3230
a_2	0.9854
τ_2 (chips)	506.1454
f_{D_2}(Hz)	-1653.8006
θ_2 (rad)	-1.2370
a_3	0.0095
τ_3 (chips)	505.5803
f_{D_3}(Hz)	-2407.6364
θ_3 (rad)	0.4319



Case 1. A table showing the output parameters (left), and the amplitude of the measured CCAF in the TEXBAT dataset (right) for 1 ms coherent integration time.

CASE 2	Output Parameters
	$\hat{g}$
a_1	1.0452
τ_1 (chips)	506.2817
f_{D_1}(Hz)	-1660.9876
θ_1 (rad)	0.1069
a_2	1.0213
τ_2 (chips)	507.8261
f_{D_2}(Hz)	-1663.4713
θ_2 (rad)	-1.5707
a_3	0.0000
τ_3 (chips)	501.5564
f_{D_3}(Hz)	-1087.3015
θ_3 (rad)	0.2369



Case 2. A table showing the output parameters (left), and the amplitude of the measured CCAF in the TEXBAT dataset (right) for 20 ms coherent integration time.

For PRN 23, as shown in Case 3, the two signals overlap very closely in code delay and it is very difficult to infer if there is another signal or just noise by looking at the CCAF magnitude alone. However, using PSO, we are still able to decompose the signals. The third output signal represents a cross-correlation peak as its magnitude is small with respect to the other two output signals.

We also tried to lower the computational load by reducing the length of z by decreasing resolution of the sample grid space $(\bar{\tau}, \bar{f}_D)$. After reducing the search space by 16 times, PSO algorithm was still able to decompose the signals into their consecutive parameters, as shown in Case 4 for PRN 13 and 1 ms coherent integration time. The results are very close to the Case 1 output parameters. The computational load for the PSO algorithm can be further reduced by finding the maximum peak using simple functions and cutting down the measurement space to ± 2 chips in code delay and ± 100 Hz in Doppler frequency.

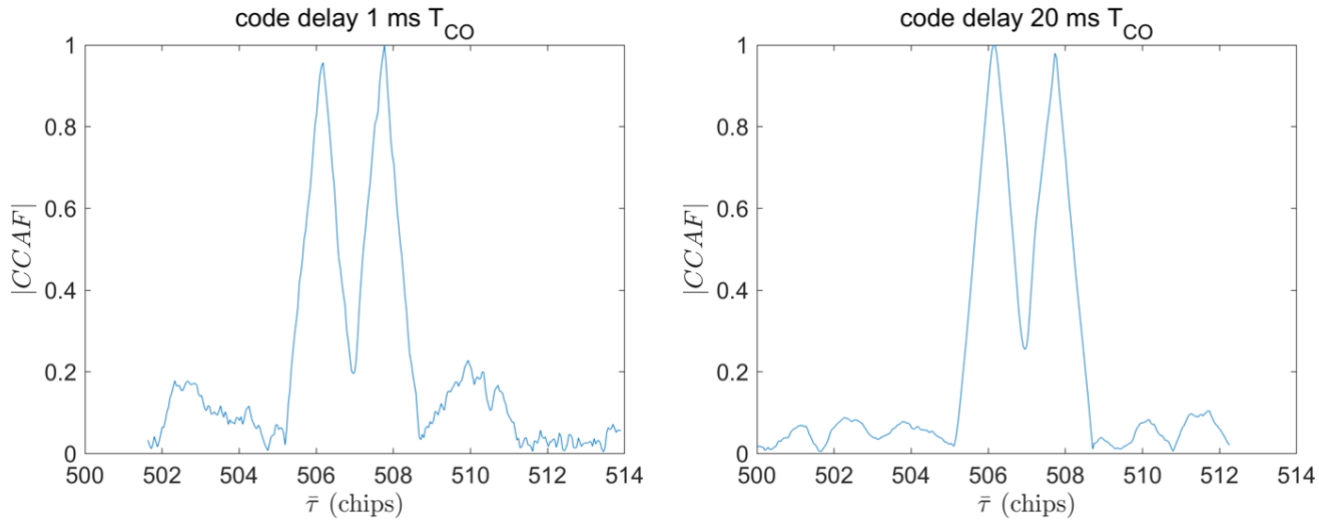
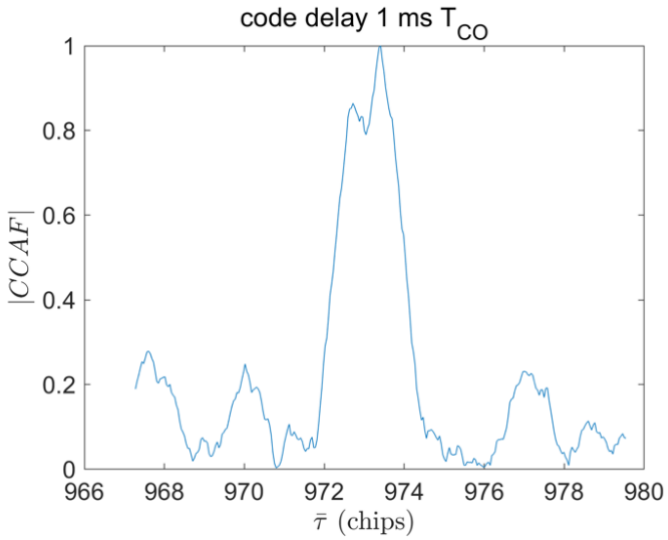


Figure 10. Constant Doppler Cut (Zoomed-in View). Code delay shows two distinct peaks (authentic and spoofed signals) with 1 ms coherent integration time (left) and 20 ms coherent integration time (right)

CASE 3	**Output Parameters**
	$\hat{g}$
$\boldsymbol{a_1}$	1.0462
$\boldsymbol{\tau_1}$ **(chips)**	973.4456
$\boldsymbol{f_{D_1}}$**(Hz)**	647.9093
$\boldsymbol{\theta_1}$ **(rad)**	-0.24512
$\boldsymbol{a_2}$	0.87985
$\boldsymbol{\tau_2}$ **(chips)**	972.7312
$\boldsymbol{f_{D_2}}$**(Hz)**	648.9084
$\boldsymbol{\theta_2}$ **(rad)**	1.5358
$\boldsymbol{a_3}$	0.1134
$\boldsymbol{\tau_3}$ **(chips)**	975.3697
$\boldsymbol{f_{D_3}}$**(Hz)**	173.0404
$\boldsymbol{\theta_3}$ **(rad)**	-0.0032682

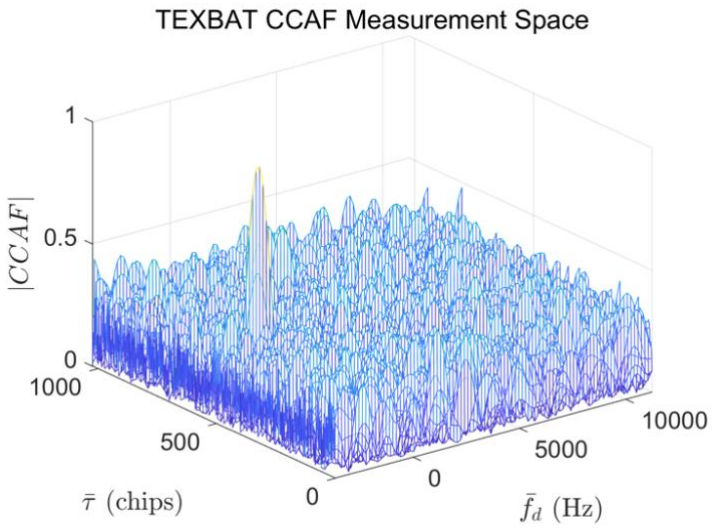


Case 3. A table showing the output parameters (left), and the amplitude of the measured CCAF in the TEXBAT dataset (right) when signals overlap very closely

PSEUDORANGE CALCULATIONS

To calculate pseudoranges, signal travel time (time between when satellites transmit the signal and when receiver receives it) must be measured. At the speed of light, a signal from a satellite at zenith would take 67 milliseconds to reach the receiver, and a signal from a satellite on the horizon would take 86 milliseconds to reach the receiver. The satellite transmittal time is sent in the navigation message in the form of Z-count, which is an increment of 1.5 seconds and specified at the beginning of each subframe. Upon receiving the signal there is uncertainty in the data bit timing. Thus, bit synchronization is done. Bit synchronization is used to find the time in a sequence where bit transitions occur. First, a zero crossing is detected. A zero crossing is where the output changes from 1 to −1, or vice versa. When a zero crossing is located, the time of a bit transition is located. When the time of one bit transition is known, it is possible to find all bit transition times since navigation bits change every 20 milliseconds with 1 millisecond epoch period.

CASE 4	Output Parameters
	$\hat{g}$
a_1	0.9760
τ_1 (chips)	507.7256
f_{D_1}(Hz)	-1627.5303
θ_1 (rad)	-0.1195
a_2	0.9464
τ_2 (chips)	506.1178
f_{D_2}(Hz)	-1661.5159
θ_2 (rad)	-1.1843
a_3	0.3640
τ_3 (chips)	509.5136
f_{D_3}(Hz)	-1061.0783
θ_3 (rad)	1.3460



Case 4. A table showing the output parameters (left), and the amplitude of the measured CCAF after decreasing resolution of the sample grid space ($\bar{\tau}, \bar{f}_D$) by 16 times.

Navigation data is modulated on the carrier wave at 50 Hz rate. The complete navigation message is placed over 25 frames, each 30 seconds long and contains 1500 bits. Each frame then contains 5 subframes. One subframe is 6 seconds long and at the beginning of each subframe there is an 8-bit preamble of 10001011, which could be inverted to 01110100 because of sign ambiguity. After finding the subframe start for all visible satellites, the times are synchronized to calculate the position using pseudoranges. The pseudorange is the signal time travel scaled by the speed of light in a vacuum. The navigation message allows us to compute satellite positions from orbital elements at the time of transmission. The pseudoranges contain unknown receiver and satellite clock biases. Satellite clock bias is estimated by the coefficients of a polynomial transmitted in the navigation message. Considering these clock biases and ionospheric and tropospheric delays, measured pseudorange *P* can be modeled as:

$$P^k = \rho^k + c(dt_i - dt^k) + T^k + I^k + e^k \tag{13}$$

$$P^k = \sqrt{(X^k - X)^2 + (Y^k - Y)^2 + (Z^k - Z)^2} + c(dt - dt^k) + T^k + I^k + e^k \tag{14}$$

where:

k	subscripts represent the satellites
ρ	is the true range
(X, Y, Z, dt)	are the position and clock bias of the receiver
(X^k, Y^k, Z^k, dt^k)	are the position and clock bias of satellite k
c	is the speed of light in a vacuum
T	is the tropospheric delay
I	is the ionospheric delay
e	is the error in measurement

After linearization, from Equation (14) the linearized pseudorange measurements (z) can be expressed as,

$$P^k - \rho^k = \begin{bmatrix} -\frac{X^1 - X}{\rho^1} & -\frac{Y^1 - Y}{\rho^1} & -\frac{Z^1 - Z}{\rho^1} & 1 \\ -\frac{X^2 - X}{\rho^2} & -\frac{Y^2 - Y}{\rho^2} & -\frac{Z^2 - Z}{\rho^2} & 1 \\ & \vdots & & \\ -\frac{X^m - X}{\rho^m} & -\frac{Y^m - Y}{\rho^m} & -\frac{Z^m - Z}{\rho^m} & 1 \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \\ cdt \end{bmatrix} - cdt^k + T^k + I^k + e^k \quad (15)$$

$$z^* = H\Delta x + \upsilon \quad (16)$$

$$\Delta\hat{x} = (H^T H)^{-1} H^T z^* \text{ (LSQ estimate)} \quad (17)$$

Direct Position Estimation

The direct position estimation approach (DPE), unlike the conventional approach, provides a navigation solution in a single step. DPE does not require a tracking loop to estimate code delay or Doppler to infer the associated PVT. The DPE approach directly estimates PVT from the received signal.

Since we do not have information about the true signal parameters, to check the accuracy of the decomposed signal parameters of all available PRNs, we estimate the position through least square position estimation. To demonstate position estimation, a clean (spoof-free) section of the TEXBAT dataset [9] is decomposed using particle swarm decomposition. The output signal parameters $\hat{\boldsymbol{g}}$ are shown in Table 1 and the corresponding estimated position is shown in Figure 11 with a red marker, which is very close to the true position identified in [12].

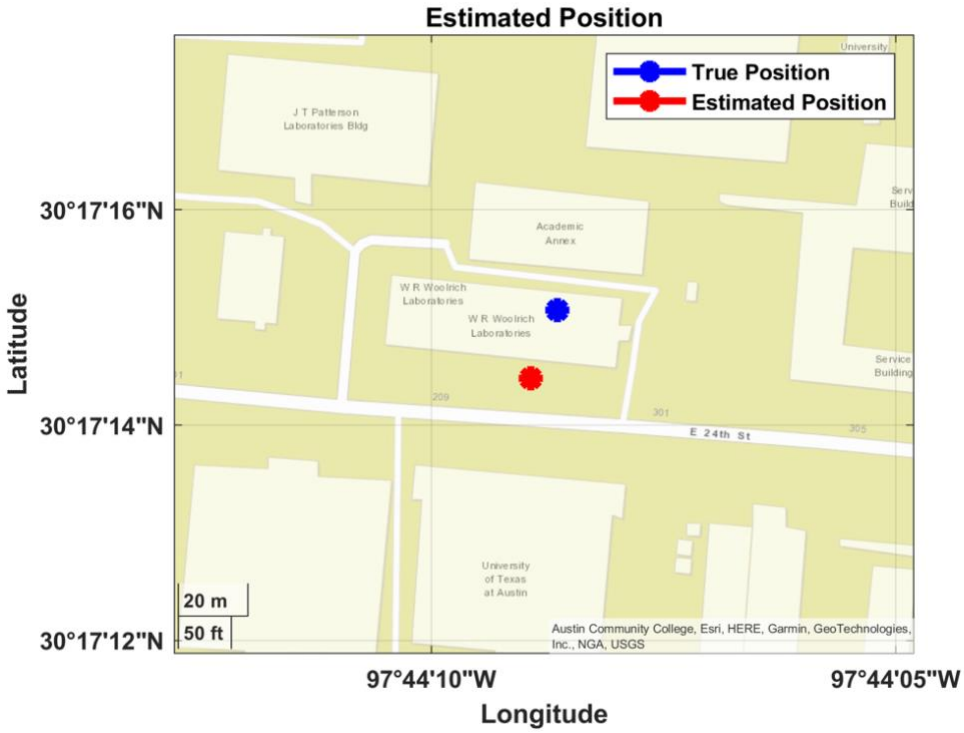


Figure 11. Estimated position with red marker in comparison with true position (30°17’15.068” N, 97°44’08.642” W) with blue marker.

PRN	Output Parameters ($\hat{g}$)			
	$\hat{a}$	$\hat{\theta}$ (rad)	$\hat{\tau}$ (chips)	$\hat{f}_D$(Hz)
23	1.0653	0.0128	756.558	693.63
13	1.0791	-0.0669	295.0373	-1637.65
3	1.0107	-0.0114	414.808	-526.06
7	1.0681	-0.0792	806.6377	-1903.02
6	1.0395	0.0033	821.1598	745.48
16	0.9935	0.0141	945.6315	2842.46

Table 1. A table showing the output decomposed parameters from different PRNs in the clean TEXBAT dataset.

Inverse receiver autonomous integrity monitoring

Receiver autonomous integrity monitoring (RAIM) is used in GNSS receivers to assess the integrity of the signals received at any instant in time. RAIM detects faults with redundant GPS pseudorange measurements. That is, when more satellites are available than needed to produce a position fix, the extra redundancy provides a measure of the measurement consistency. For example, in residual-based RAIM, the test statistic is defined as the 2-norm of the residual vector *r* (i.e., the 2-norm difference between the estimated and observed measurements):

$$r \triangleq z^* - H\Delta\hat{x} \tag{18}$$

For spoofing detection, 6 satellite signals are decomposed into 3 signals each, resulting in *n* combinations of 6 satellites per set. In Figure 12, different satellites are represented with different colors. Using PSO, each of the satellites produces 3 outputs signals (authentic, spoofed, multipath; represented as 1, 2, and 3, respectively). Each set provides a position fix. If the sets contain all authentic and all spoofed signals, the residual r will be small illustrating a consistency among the signals in the set, while other combination sets will not. Since we are interested in consistent sets, instead of inconsistent, we dub this process "Inverse RAIM". In Figure 13, we plot all the position fixes (red markers) with the true position shown as a blue marker. In Figure 14 (left), all 64 combination sets' estimated positions are plotted by set number. As shown in Figure 14 (right), the residuals for all combinations are also shown. Note that the residual of four combination sets, numbered 1, 5, 60, 64, are smaller in comparison to the rest. In these results, satellite combination set 1 is the conjugate of satellite combination Set 64, which means that if one combination contains all the code phases from the first peak, the other combination contains code phases from the second peak.

PRN	Output code phase $\hat{\tau}$ (chips)	
	First peak	Second peak
23	965.1516	964.9815
13	526.7222	525.1371
3	635.2136	634.5055
7	17.5546	16.0155
6	6.3839	5.3196
16	110.0041	108.2743

Table 2. A table showing the output decomposed parameters from different PRNs in TEXBAT dataset spoofing Scenario 4 (power matched position push).

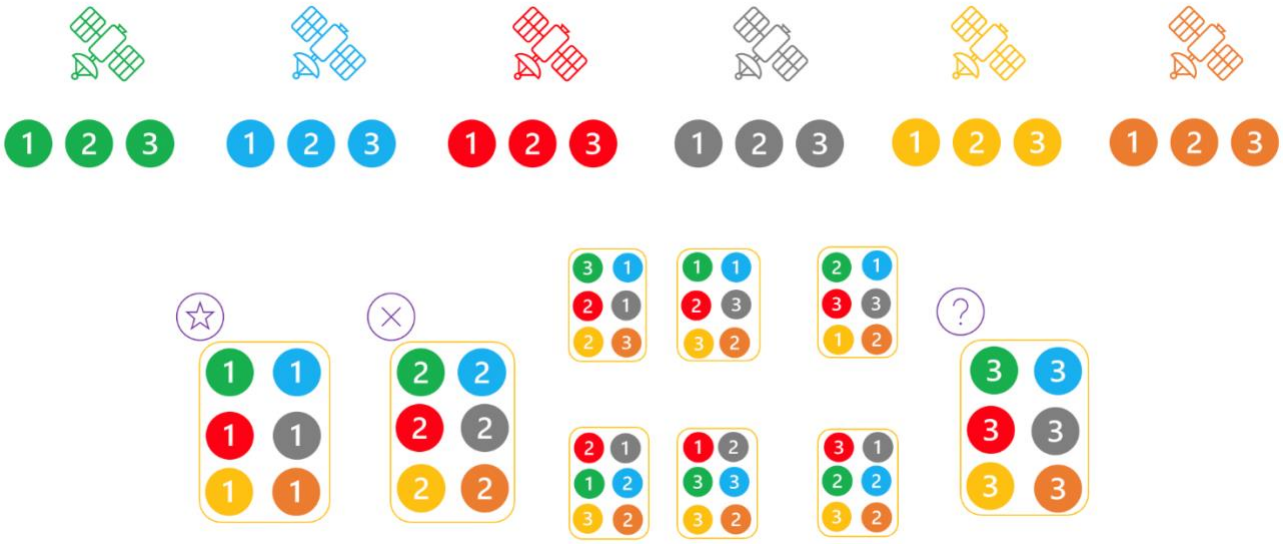


Figure 12. Inverse receiver autonomous integrity monitoring concept with three decomposed signals from each satellite numbered as: (1) authentic signal, (2) spoofed signal, (3) multipath.

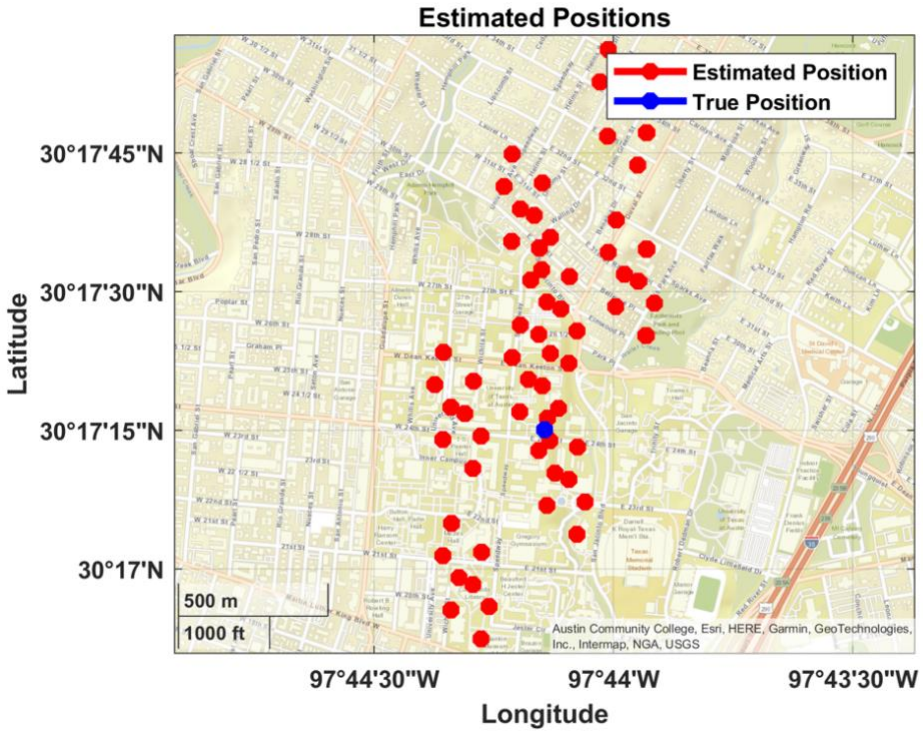


Figure 13. Estimated positions from different sets of satellite combination (authentic and spoofed) shown with red markers in comparison with true position (30°17'15.068" N, 97°44'08.642" W) with blue marker.

In Figure 15, we show the estimated position error for 5 timestamps spanning over 100 seconds with 20 second intervals, for the 4 identified sets. This figure illustrates that combination set 64 has the least error in all three directions, hence likely representing the true position. As the TEXBAT paper [9] indicates for Scenario 4 (position push spoofing attack), the victim position changes 600 meters (equivalent to 2 chips) in the vertical direction, which seems to match set 1, albeit with a somewhat different magnitude than claimed in [7]. That, and given that Set 1 is the conjugate of combination of Set 64, shows that Set 1 is the spoofing signal set. Although in this analysis, we utilized the prior knowledge that TEXTBAT data corresponds to a

static user in differentiating between the authentic and spoofed signals, such information about the user dynamics can also be realistically achieved using other aiding sensors (for example, inertial sensors). With such aiding, spoofing attacks can not only be detected, but also mitigated by forcing the receiver to output the position estimate corresponding to the authentic combination set out of the decomposed signals.

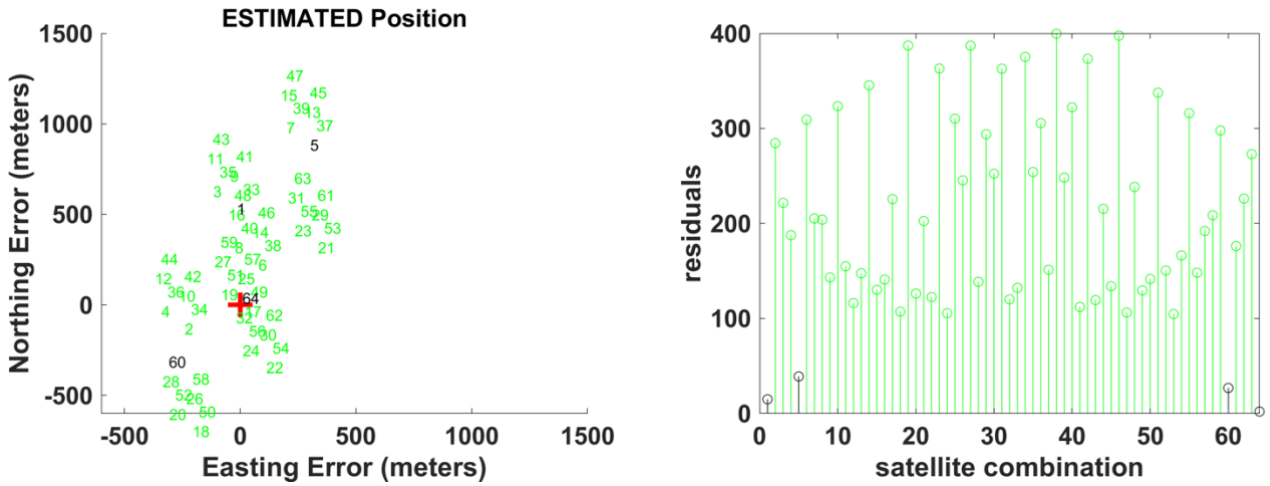


Figure 14. Position fixes for 64 satellite combination sets (left) with red marker as true position, and residuals corresponding to those combination sets (right).

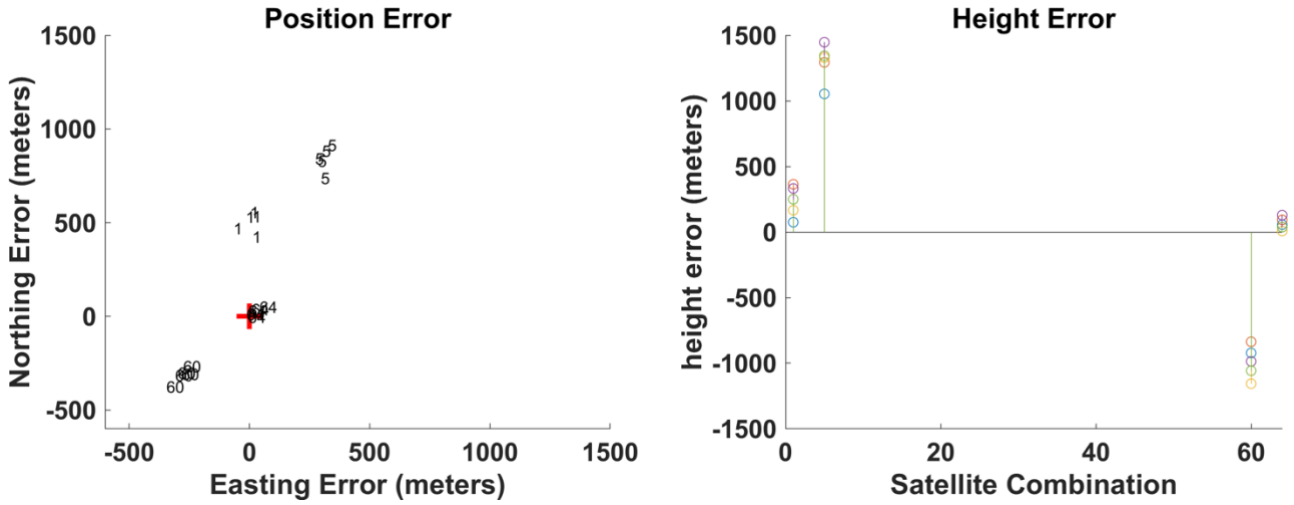


Figure 15. Position fixes for 4 satellite combination sets with minimum residuals (left) with red marker as true position, and residuals corresponding to those combination sets (right) over 100 seconds with 20 second time intervals.

Figure 16 shows the results of decomposition and direct position estimation over 3 consecutive milliseconds. The actual positions should be constant over such a short time interval, but the exhibited variations in the estimated positions are still rather large. We expect that increasing the coherent integration time in future work to 20 ms will help to reduce the error further.

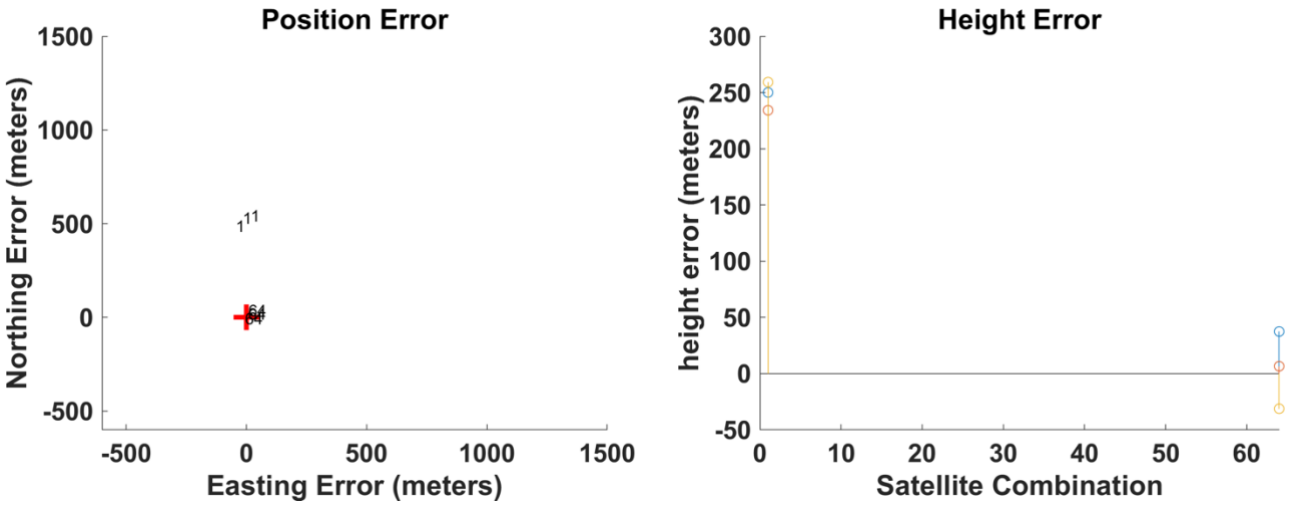


Figure 16. Estimated positions from different sets of satellite combinations (authentic and spoofed) shown with red markers in comparison with true position (30°17'15.068" N, 97°44'08.642" W) with blue marker.

CONCLUSION

In this paper, we have shown a method for CCAF decomposition integrated with direct positioning and inverse RAIM for detecting and mitigating spoofing attacks. This method decomposes the CCAF into three signals (authentic, spoofed, and multipath) and estimates the output component parameter vector $\hat{g}$. Decomposed output parameters are used for direct position estimation by combining different combination sets. We suggest a spoofing monitor based on pseudorange residual errors. Out of all the combination sets, only two will be consistent in a RAIM sense: when all the authentic signals from each PRN are together in one set, and when all the spoofed signals from each PRN are together in another. We also demonstrated how the decomposition of the signals allowed continuous tracking and estimation of the true position. Future efforts will include increasing the coherent integration time to get better estimates of code phases and implementing measurement correlation in the decomposition cost function to reduce the error. Also, integrating inertial sensors with CCAF decomposition and inverse RAIM will mitigate spoofing attacks even for dynamic users.

REFERENCES

[1] S. Ahmed, S. Khanafseh and B. Pervan, "GNSS Spoofing Detection based on Decomposition of the Complex Cross Ambiguity Function," in *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, St. Louis, Missouri, 2021.

[2] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. O'Hanlon and P. M. Kintner, "Assessing the Spoofing Threat : Development of a Portable GPS Civilian Spoofer," in *Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008)*, Savannah GA, 2008.

[3] C. Tanil, "Detecting GNSS Spoofing Attacks Using INS Coupling," in Ph.D. Dissertation, Department of Mechanical and Aerospace Engineering, Illinois Institute of Technology, Chicago, IL, 2016.

[4] B. Kujur, S. Khanfseh and B. Pervan, "A Solution Separation Monitor using INS for Detecting GNSS Spoofing," in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020), September 2020, pp. 3210-3226.*.

[5] M. Pini, M. Fantino, A. Cavaleri, S. Ugazio and L. L. Presti, "Signal Quality Monitoring Applied to Spoofing Detection," in *Proceedings of the 24th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2011)*, Portland OR, 2011.

[6] E. G. Manfredini, D. M. Akos, Y.-H. Chen, S. Lo, T. Walter and P. Enge, "Effective GPS Spoofing Detection Utilizing Metrics from Commercial Receivers," in *Proceedings of the 2018 International Technical Meeting of The Institute of Navigation*, Reston, Virginia, 2018.

[7] H. Christopher,, B. O'Hanlon, A. Odeh, K. Shallberg and J. Flake, "Spoofing Detection in GNSS Receivers through CrossAmbiguity Function Monitoring," in *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, Miami, Florida, 2019.

[8] P. Borhani-Darian, H. Li, P. Wu and P. Closas, "Deep Neural Network Approach to Detect GNSS Spoofing Attacks," in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*.

[9] T. Humphreys, J. Bhatti, D. Shepard and K. Wesson, "The Texas Spoofing Test Battery: Toward a Standard for Evaluating GPS Signal Authentication Techniques," in *Proceedings of the 25th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2012)*, Nashville, TN, 2012.

[10] K. Borre, D. Akos, N. Bertelsen, P. Rinder and S. H. Jensen, A Software-Defined GPS and Galileo Receiver:A Single-Frequency Approach, Boston, MA: Birkhäuser .

[11] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks, 1995, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968*.

[12] A. Lemmenes, P. Corbell and S. Gunawardena, "Detailed Analysis of the TEXBAT Datasets Using a High Fidelity Software GPS Receiver," in *Proceedings of the 29th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2016)*, Portland, Oregon, September 2016.

[13] M. Foucras, J. Leclère, C. Botteron, O. Julien, C. Macabiau, P.-A. Farine and B. Ekambi, "Study on the cross-correlation of GNSS signals and typical approximations," in *GPS Solutions, Springer Verlag*, 2017.

[14] K. D. Wesson, D. P. Shepard, J. A. Bhatti and T. E. Humphreys,, "An Evaluation of the Vestigial Signal Defense for Civil GPS Anti-Spoofing," in *Proceedings of the 24th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2011)*, Portland, OR, 2011.

APPENDIX

Here we derived the approximation in equation 4 from the expression given:

$$S = \frac{\sqrt{C}}{T_{CO}} \int_0^{T_{CO}} x(t)x(t-\tau)\exp\left(2\pi(f_D - \bar{f}_D)t + \theta - \bar{\theta}\right) dt \tag{19}$$

Taking the expected value of the signal and not taking amplitude and phase term into consideration as it is time invariant

$$S = E\left\{\frac{1}{T_{CO}} \int_0^{T_{CO}} x(t)x(t-\tau)\exp(j2\pi\Delta f_D t)\, dt\right\} \tag{20}$$

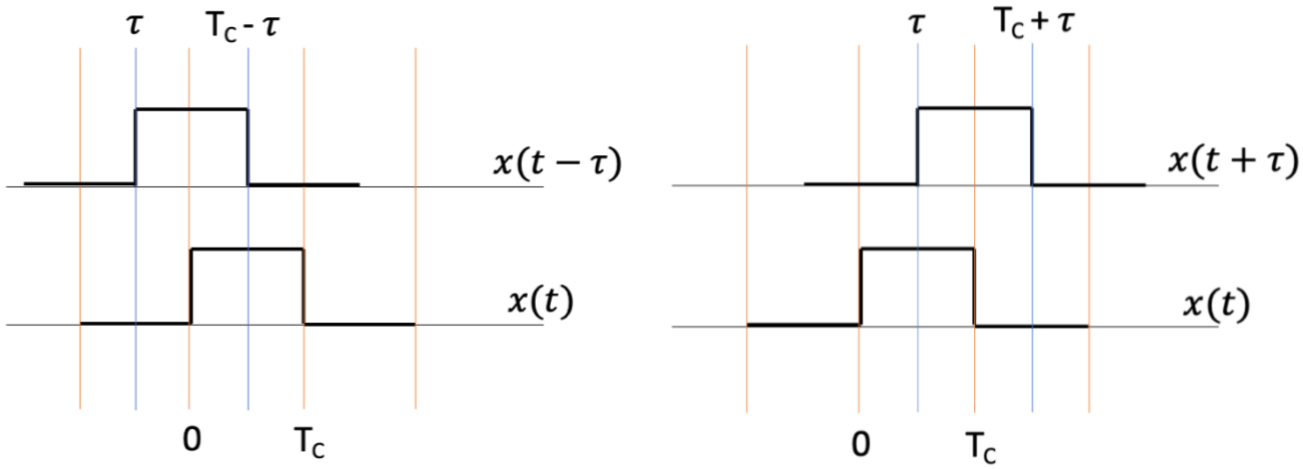


Figure A. Case where incoming code chip misaligned with local code chip early (left) and late (right)

Case 1 : $-T_C < \tau < 0$

When incoming code delay is received early with respect to the local generated code

$$S = \frac{1}{T_{CO}} \sum_{n=0}^{N-1} \int_{nT_C}^{(n+1)T_C+\tau} \exp(j2\pi\Delta f_D t)\, dt \tag{21}$$

where:
T_C is the duration of one chip
N is the number of code periods

And,

$$T_{CO} = NT_C \tag{22}$$

$$S = \frac{1}{j2\pi f_D} \frac{1}{T_{CO}} \sum_{n=0}^{N-1} \{\exp(j2\pi\Delta f_D((n+1)T_C + \tau)) - \exp(j2\pi\Delta f_D nT_C) \tag{23}$$

$$S = \frac{1}{j2\pi f_D T_{CO}} [\exp(j2\pi\Delta f_D(T_C + \tau) - 1] \sum_{n=0}^{N-1} \exp(j2\pi\Delta f_D nT_C) \tag{24}$$

Assume, $\alpha \cong \exp(j2\pi\Delta f_D T_C)$

$$\mathrm{c} = 1 + \alpha + \alpha^2 + \cdots + \alpha^{N-1} \tag{25}$$

$$\alpha\mathrm{c} = \alpha + \alpha^2 + \cdots + \alpha^{N-1} + \alpha^N \tag{26}$$

$$(1-\alpha)\mathrm{c} = 1 - \alpha^N \tag{27}$$

$$c = \frac{1-\alpha^N}{1-\alpha} \tag{28}$$

$$c = \frac{1 - \exp(j2\pi\Delta f_D NT_C)}{1 - \exp(j2\pi\Delta f_D T_C)} \tag{29}$$

$$c = \frac{\exp(-j\pi\Delta f_D T_{CO}) - \exp(j\pi\Delta f_D T_{CO})}{\exp(-j\pi\Delta f_D T_C) - \exp(j\pi\Delta f_D T_C)} \frac{\exp(j\pi\Delta f_D T_{CO})}{\exp(j\pi\Delta f_D T_C)} \tag{30}$$

$$c = \frac{\sin(\pi\Delta f_D T_{CO})}{\sin(\pi\Delta f_D T_C)} \exp(j\pi\Delta f_D(N-1)T_C) \tag{31}$$

$$S = \frac{1}{j2\pi f_D T_{CO}} [\exp(j2\pi\Delta f_D(T_C + \tau) - 1] \frac{\sin(\pi\Delta f_D T_{CO})}{\sin(\pi\Delta f_D T_C)} \exp(j\pi\Delta f_D(N-1)T_C) \tag{32}$$

$$S = \frac{1}{j2\pi\Delta f_D T_{CO}} [\cos(2\pi\Delta f_D(T_C + \tau) - 1 + j\sin(2\pi\Delta f_D(T_C + \tau)] \frac{\sin(\pi\Delta f_D T_{CO})}{\sin(\pi\Delta f_D T_C)} \exp(j\pi\Delta f_D(N-1)T_C) \tag{33}$$

$$S = \frac{1}{j2\pi\Delta f_D T_{CO}}[-2\sin^2(\pi\Delta f_D(T_C+\tau) + j2\sin(\pi\Delta f_D(T_C+\tau)\cos(\pi\Delta f_D(T_C+\tau)]\ \frac{\sin(\pi\Delta f_D T_{CO})}{\sin(\pi\Delta f_D T_C)}\exp(j\pi\Delta f_D(N-1)T_C) \quad (34)$$

$$S = \frac{1}{j\pi\Delta f_D T_{CO}}\sin(\pi\Delta f_D(T_C+\tau)\sin(\pi\Delta f_D T_{CO})[-\sin(\pi\Delta f_D(T_C+\tau) + j\cos(\pi\Delta f_D(T_C+\tau)]\ \frac{1}{\sin(\pi\Delta f_D T_C)}\exp(j\pi\Delta f_D(N-1)T_C) \quad (35)$$

Using identities

$\frac{1}{j} = \exp\left(-j\frac{\pi}{2}\right)$ and $-\sin(\phi) + j\cos(\phi) = \cos(\frac{\pi}{2}+\phi) + j\sin(\frac{\pi}{2}+\phi)$

$$S = \exp\left(-j\frac{\pi}{2}\right)\text{sinc}(\pi\Delta f_D T_{CO})\ \frac{\sin(\pi\Delta f_D(T_C+\tau))}{\sin(\pi\Delta f_D T_C)}\ \exp\left(j\frac{\pi}{2}+j\pi\Delta f_D(T_C+\tau)\right)\exp(j\pi\Delta f_D(N-1)T_C) \quad (36)$$

$$S = \text{sinc}(\pi\Delta f_D T_{CO})\ \frac{\sin(\pi\Delta f_D(T_C+\tau))}{\sin(\pi\Delta f_D T_C)}\exp(j\pi\Delta f_D(T_{CO}+\tau)) \quad (37)$$

Case 2 : $0 < \tau < T_{CO}$

When incoming code delay is received late with respect to the local generated code

$$S = \frac{1}{T_{CO}}\sum_{n=0}^{N-1}\int_{nT_C+\tau}^{(n+1)T_C} x(t)x(t-\tau)\exp(j2\pi\Delta f_D t)\,dt \quad (38)$$

$$S = \frac{1}{j2\pi\Delta f_D}\frac{1}{T_{CO}}\sum_{n=0}^{N-1} x(t)x(t-\tau)\{\exp(j2\pi\Delta f_D(n+1)T_C) - \exp(j2\pi\Delta f_D(nT_C+\tau)) \quad (39)$$

$$S = \frac{1}{j2\pi\Delta f_D T_{CO}}[\exp(j2\pi\Delta f_D T_C) - \exp(j2\pi\Delta f_D\tau)]\sum_{n=0}^{N-1} x(t)x(t-\tau)\exp(j2\pi\Delta f_D nT_C) \quad (30)$$

From (31), using the expression we get

$$S = \frac{1}{j2\pi\Delta f_D T_{CO}}[\exp(j2\pi\Delta f_D T_C) - \exp(j2\pi\Delta f_D\tau)]\ \frac{\sin(\pi\Delta f_D T_{CO})}{\sin(\pi\Delta f_D T_C)}\exp(j\pi\Delta f_D(N-1)T_C) \quad (41)$$

$$S = \frac{1}{j2\pi\Delta f_D T_{CO}}[\cos(2\pi\Delta f_D T_C) - \cos(2\pi\Delta f_D\tau) + j[\sin(2\pi\Delta f_D T_C) - \sin(2\pi\Delta f_D\tau)]]\ \frac{\sin(\pi\Delta f_D T_{CO})}{\sin(\pi\Delta f_D T_C)}\exp(j\pi f_D(N-1)T_C) \quad (42)$$

$$S = \frac{1}{j2\pi\Delta f_D T_{CO}}[-2\sin(\pi\Delta f_D(T_C+\tau))\sin(2\pi\Delta f_D(T_C-\tau)) + j2\cos(\pi\Delta f_D(T_C+\tau))\sin(2\pi\Delta f_D(T_C-\tau))]\ \frac{\sin(\pi\Delta f_D T_{CO})}{\sin(\pi\Delta f_D T_C)}\exp(j\pi\Delta f_D(N-1)T_C) \quad (43)$$

$$S = \exp\left(-j\frac{\pi}{2}\right)\mathrm{sinc}(\,\pi\Delta f_D T_{CO})\;\frac{\sin(\pi\Delta f_D(T_C-\tau))}{\sin(\,\pi\Delta f_D T_C)}\;\exp\left(j\frac{\pi}{2}+j\pi\Delta f_D(T_C+\tau)\right)\exp(j\pi\Delta f_D(N-1)T_C) \tag{44}$$

$$S = \mathrm{sinc}(\,\pi\Delta f_D T_{CO})\;\frac{\sin(\pi\Delta f_D(T_C-\tau))}{\sin(\,\pi\Delta f_D T_C)}\exp(j\pi\Delta f_D(T_{CO}+\tau)) \tag{45}$$

Combining equation (37) and (45), we get

$$S = \begin{cases} \mathrm{sinc}(\,\pi\Delta f_D T_{CO})\;\dfrac{\sin(\pi\Delta f_D(T_C-|\tau|))}{\sin(\,\pi\Delta f_D T_C)}\exp(j\pi\Delta f_D(T_{CO}+\tau)) & -T_c<\tau<T_c \\ 0 & \text{otherwise} \end{cases} \tag{46}$$

$$S = \begin{cases} \mathrm{sinc}(\,\pi\Delta f_D T_{CO})\;\dfrac{\sin(\pi\Delta f_D(T_C-|\tau|))}{\sin(\,\pi\Delta f_D T_C)}\exp\left(j\pi\Delta f_D T_{CO}(1+\dfrac{\tau}{T_{CO}})\right) & -T_c<\tau<T_c \\ 0 & \text{otherwise} \end{cases} \tag{47}$$

Since,

$$|\tau| < T_c \quad \& \quad \tfrac{\tau}{T_{CO}} < 10^{-3} \ll 1$$

And,

$$\lim_{f\to 0}\frac{\sin(\pi\Delta f_D(T_C-|\tau|))}{\sin(\,\pi\Delta f_D T_C)} = (1-\frac{|\tau|}{T_C})$$

$$S = \begin{cases} \left(1-\dfrac{|\tau|}{T_C}\right)\mathrm{sinc}(\,\pi\Delta f_D T_{CO})\;\exp(j\pi\Delta f_D T_{CO}) & -T_c<\tau<T_c \\ 0 & \text{otherwise} \end{cases} \tag{48}$$

Finally, the approximation is written as

$$S\left(\sqrt{C},\tau,f_D,\theta;\bar{\tau},\,\bar{f}_D,\bar{\theta}\right) = \sqrt{C}\;R(\tau-\bar{\tau})\;\mathrm{sinc}\left(\pi(f_D-\bar{f}_D)T_{CO}\right)\exp\left(i\pi((f_D-\bar{f}_D)T_{CO}+\theta-\bar{\theta})\right)$$